

**IMPLEMENTATION OF AN ALGORITHM FOR CYLINDRICAL OBJECT  
IDENTIFICATION USING RANGE DATA**

Sylvia T. Bozeman and Benjamin J. Martin  
Department of Mathematics  
Spelman College  
Atlanta, Georgia

512-61

26593

P.S.

**ABSTRACT**

This research addresses one of the problems in 3-D object identification and localization. In robotic and navigation applications the vision system must be able to distinguish cylindrical or spherical objects as well as those of other geometric shapes. An algorithm has been developed to identify cylindrical objects in an image when range data is used. The algorithm incorporates the Hough transform for line detection using edge points which emerge from a Sobel mask. Slices of the data are examined to locate arcs of circles using the normal equations of an over-determined linear system. Current efforts are devoted to testing the computer implementation of the algorithm. Refinements are expected to continue in order to accommodate cylinders in various positions. A technique is sought which is robust in the presence of noise and partial occlusions.

# IMPLEMENTATION OF AN ALGORITHM FOR CYLINDRICAL OBJECT IDENTIFICATION USING RANGE DATA

## Introduction

Among the various tasks assigned to the National Aeronautics and Space Administration is the investigation of space related automation and robotics. NASA has chosen to use the telerobotic system configuration as the basis for progressively studying and implementing spaceborne automation. Telerobotic systems consist of a human, a robot, a human-robot interface capability, sensors, and the accompanying computational tools. By basing research on this system structure, NASA can progressively evolve its automation capability from teleoperator, where the human is directly involved in every operation of the machine, to near autonomy, where man is only involved as a monitor of the machine's activity.

In designing automated robotic activity, it is necessary to be able to automatically (i.e., without human involvement) recognize objects, their positions, and their orientations in order to be able to grasp and manipulate them. Thus the ability to take the data from the scanning rangefinders and determine the composition of the image scanned and the locations of the various components is needed. Even as work progresses on the development of faster and more accurate scanners, considerable research is underway on methods for using range data in object detections and recognition. (See reference list of Faugeras and Herbert, 1986.)

In the problem of 3-D recognition and localization of objects that are rigid, a major consideration is the method of representing objects. After considerable investigation Faugeras and Herbert (1986) concluded that representation should be in terms of linear primitives, such as points, lines, and planes, although these may be derived from consideration of curved patches at intermediate stages.

The technique here employed for detecting cylinders is based on the use of external descriptors. In the case of the cylinder it is noted that in one direction the boundary should appear as straight lines. In a direction orthogonal to the first the data should fit the equation of a circle (or an ellipse in the more general case). Information on the boundary in the third orthogonal direction is not used.

The approach used in this project has been to develop the algorithm while testing it on ideal data for a right circular cylinder with the axis of the cylinder parallel to the y-axis. This condition is later relaxed to allow the introduction of an angle  $\alpha$  that indicates the angle between the cylinder's axis and the y-axis; thus freedom of rotation in one direction is allowed. The ultimate goal is to refine the algorithm so that it is robust in noisy environments. The algorithm should also yield a high rate of accuracy when the cylinders have partial occlusions.

In approaching this problem the spirit of a comment attributed to A. Guzman is followed: "If you recognize a foot you know where (in the picture) to look for a leg..." Hence the procedure for locating cylinders in a picture has been to look first for parallel lines; then the algorithm can check for a cylinder between the lines. Therefore, the first major part of the programming project consisted of identifying edgepoints in the image and extracting lines from those edgepoints. Both tasks have been the subject of considerable research over the past 20 years. The basic methods employed in this part of the algorithm are then somewhat classical in nature.

## Locating Edgepoints and Lines

The first step is to locate the pixels that are the edgepoints. At each pixel in the image a 3x3 mask centered at (i, j) is used to determine the variation on distance from the sensor in both the vertical and the horizontal directions. Note that i represents the row, j the column, and P(i, j) the range data at the pixel location (i, j). This mask is known as a basic Sobel edge detector where

$$\text{igrad} = 2(P(i-1,j)-P(i+1,j)) + P(i-1,j+1)-P(i+1,j+1) + P(i-1,j-1)-P(i+1,j-1)$$

$$\text{jgrad} = 2(P(i,j+1)-P(i,j-1)) + P(i-1,j+1)-P(i-1,j-1) + P(i+1,j+1)-P(i+1,j-1)$$

estimate the variations in the vertical and horizontal directions, respectively, at (i,j). Using these two quantities in a vector  $G=(\text{jgrad}, \text{igrad})$  to approximate the gradient vector at (i,j), both the magnitude and direction of the greatest variation at (i,j) can be obtained. If the magnitude of the gradient,

$$k = ((\text{igrad})^2 + (\text{jgrad})^2)^{1/2},$$

is above some predetermined threshold value, T, then it is assumed that there is an edgepoint at (i,j). T may vary and can be selected interactively prior to runs. Using a threshold of 100 it was found that certain edgepoints around the top of the cylinder were not detected even with ideal data.

In searching for lines among the edgepoints, the gradient directional information may or may not be used. Using gradient information, the edge angle at (i,j) is found as  $g = \arctan(\text{igrad}, \text{jgrad})$ .

J. Kittler [1983] has given considerable attention to the performance of the Sobel edge detector in terms of the accuracy of its magnitude and angle. He proved that the basic Sobel edge detector described above will correctly detect the true angle of the edge direction at the boundary between the object and the background in the interval  $0 \leq g \leq \arctan(\frac{1}{3})$ . He predicts an error of approximately  $2^\circ$  in direction for larger angles and an accurate magnitude only when the edge is vertical or horizontal. Davis [1987] points out that this technique loses accuracy when the edges of the objects are fuzzy.

In attempting to extract lines from among the edgepoints the Hough transform has been shown to be an effective technique. It is especially attractive in this environment because of its robust performance in noisy environments. Although it can be employed with or without the gradient information, we chose to use the gradient information in order to reduce the number of lines associated with each pixel and thereby reduce the storage requirements.

This technique for detecting collinear points dates back to 1962 with a patent issued to P.V.C. Hough. The technique is based on the fact that a line through a point (x,y) can be completely characterized by its angle (from the positive x-axis) and the length of the normal from the origin to the line. The line can be expressed using the equation

$$x_i \cos \theta + y_i \sin \theta = r. \quad (1)$$

This approach to finding lines is based on the fact that if enough pairs  $(x_i, y_i)$  appear in the image which give rise to the same  $(r, \theta)$  pair then these points may well represent a line in the image (if they can be shown to be contiguous). Thus a "voting" system is established and a two-dimensional accumulator array is created to count the votes. When gradient information is not used the angle  $\theta$  is allowed to vary independently over a  $180^\circ$  range (using small increments) and each point  $(x_i, y_i)$  can produce a value of  $r$  for each value of  $\theta$ . If  $r$  lies within the limits specified by the physical dimensions of the image, the number of votes at location  $(r, \theta)$  is incremented by one. When gradient information is used at each point, the only value of  $(r, \theta)$  incremented for that point is the one where  $\theta$  corresponds to the gradient at the point, and  $r$  is calculated using (1).

In the latter case Gorman and Clowes [1976] recommended the creation of "contributing point lists" to keep track of the points whose votes account for the value in a particular cell of the accumulator array. The data structure chosen here to accomplish this tracking was a second array, matching the accumulator array, which consisted of pointers, each pointing to the list of pairs contributing to the value at  $(r, \theta)$ . Once the accumulator array is filled (i.e. once the selected portion of the image has been processed) the list would contain all points found to be on or near the line specified by the parameters  $(r, \theta)$ .

This method works well to find the line containing line segments in the image but has two shortcomings. (1) It does not give any indication as to whether the points found are contiguous. (2) It does not give the location of line segments on the lines found. The difficulty resulting from the latter is somewhat diminished by employing the contributing point lists.

## Isolating Edges and Cross-Sections of a Cylinder

Attention is now turned to analyzing the data in the parameterized space of  $(r, \theta)$ . Since finding consecutive pairs of parallel lines is of interest, peaks in the data taken from a cross-section of the space are sought. For a fixed value of  $\theta$ , consecutive values  $r_1$  and  $r_2$  are sought in which the accumulation of "votes" for a line at  $(r_i, \theta)$  exceeds some predetermined value  $M$ . The value  $M$  may be selected interactively and should depend on (1) the required length of a line in order to be considered meaningful, and (2) the size of the image (or subimage) being processed.

In an image of size 256x256 pixels a value of  $M=20$  was used. The two problems that arise with this (as with any) choice of  $M$  are that (1) cylinders of less than 20 pixels in height go undetected, and (2) if less than 20 pixels are visible along the edge of the cylinder due to occlusion by other objects in the image, then the necessary matching pair of edges will not be found. If peaks appear in the parameterized data at  $(r_1, \theta)$  and  $(r_2, \theta)$  but for no other value of  $r$ ,  $r_1 < r < r_2$ , then a cylinder is sought between the corresponding parallel lines  $l_1$  and  $l_2$ .

## Labeling in 3-D

Since a right circular cylinder in standard position (i.e., having its axis parallel to the y-axis) has uniform cross sections whose boundaries are all circles of the same radius, the goal is to isolate a cross section of the image between the two parallel lines. The points along the visible portion of the cross

section will be tested to determine if they lie on an arc of a circle—a circle which in 3-D lies perpendicular to the plane of the image.

It is only now that the labeling of the points in the image of the 3-D space becomes important. Prior to this point in the discussion, consideration of the projection of 3-D objects onto the image plane has been sufficient. If an  $xy$ -coordinate system is superimposed onto the image, so that the origin of the system lies in the lower left corner of the image (at a background point), then each pixel location  $(i,j)$  also can be referenced as  $(x,y)$ , where  $x=j$  and  $y=256-i$ . (Figure 1.)

If a third coordinate axis  $z$  which is orthogonal to the  $xy$  image plane is added then the distance from a point  $R$ , located on an object in the scene, to the plane of the sensor can be measured along  $z$ . (Figure 1.) The grey level of each pixel in the image represents the range data and indicates the relative distance of the object from the sensor—lighter areas (higher grey levels) being closer. Thus if  $z=0$  at  $R$  then the point  $R$  is as far as possible from the sensor. Let  $R$  be a given point on an object in the scene which appears at location  $(i,j)$  in the 2-D image. Suppose that the plane of the sensor is at a distance  $S$  from the reference background point at the origin of the coordinate system. If  $k$  is the 3-D distance at point  $R$  from the object to the sensor, then the  $z$ -coordinate of  $R$  is  $z=S-k$ . Thus, the point  $R$  can be positioned relative to the  $xyz$ -coordinate system since its  $x$  and  $y$  coordinates are the  $x$  and  $y$  coordinates of  $(i,j)$  (Figure 2).

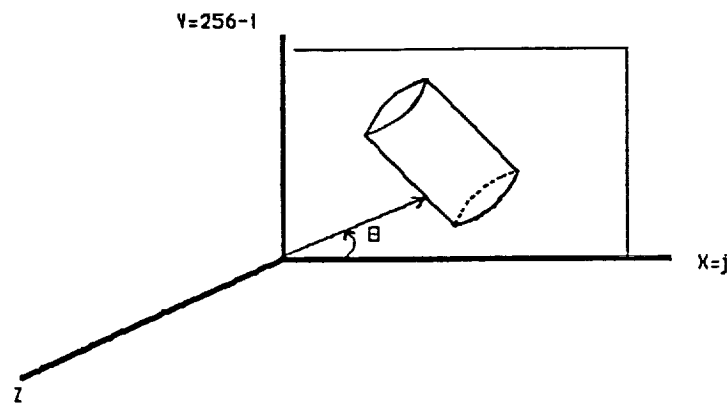


Figure 1.

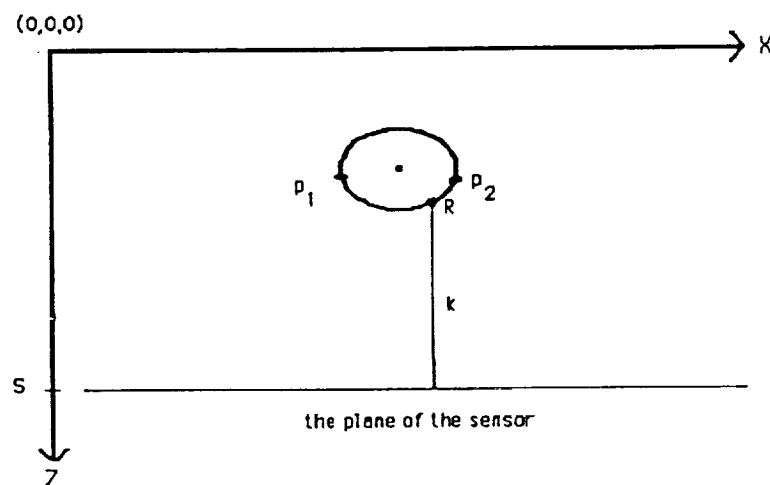


Figure 2.

## Analyzing a "Slice" of the Data

In locating data which might lie on the arc of a circle between the two parallel lines  $l_1$  and  $l_2$  already found, a point is chosen from each of the two lines which lies on a single slice of the data. Thus we seek two points  $p_1$  and  $p_2$  that are endpoints of a line segment perpendicular to  $l_1$  and  $l_2$ , so that  $p_1$  lies on  $l_1$  and  $p_2$  lies on  $l_2$ . For the cylinder in a standard position such points have the same  $y$  (or  $y$ ) value. Thus we need only return to the lists of contributing points corresponding to the  $(r, \theta)$  values giving rise to the two lines and locate two points with the same  $y$  coordinate. (The search is made easier by ordering points as the lists are created.) In the more general case, the angle of inclination of the parallel lines can be used to determine the relationship between the  $y$  values of the two corresponding points sought.

In order to determine whether  $p_1$  and  $p_2$  are the endpoints of the arc of a circle in 3-D space we test the range data at these two points and points in between (on the same orthogonal slice). Denote by  $D$  this set of points. If we confine ourselves for a moment to the case of the cylinder in standard position, it is no longer necessary to consider the  $y$  direction for points on the slice. Thus each point in the image can be considered as having only  $x$  and  $z$  components (Figure 2). In this case

$$D = \{(x, y_0, z) \mid x_1 < x < x_2, \text{ where } p_1 = (x_1, y_0, z_1), p_2 = (x_2, y_0, z_2)\}$$

In this two-dimensional subspace the equation of a circle (whose center has  $y$ -component  $y_0$ ) determined uniquely by any three points in  $D$  is sought. However, if the points of  $D$  approximate an arc all of its points must lie on, or near, the circle determined.

If we suppress the constant component, then for any point  $(x_i, z_i)$  in  $D$ ,

$$(x_i - x_c)^2 + (z_i - z_c)^2 = r^2 \quad (1)$$

if the point lies on the circle with center  $(x_c, y_0, z_c)$  and radius  $r$ . By equating two such equations using points  $(x_i, z_i)$  and  $(x_j, z_j)$  we can eliminate  $r$  and get the equation

$$2x_c (x_i - x_j) + 2z_c (z_i - z_j) = x_i^2 + z_i^2 - x_j^2 - z_j^2,$$

a linear equation in the two unknowns  $x_c$  and  $z_c$ .

Selecting  $m$  points  $(x_i, z_i)$ ,  $i=1, \dots, m$ , from  $D$ ,  $m-1$  equations in 2 unknowns can be formed as

$$2x_c (x_i - x_m) + 2z_c (z_i - z_m) = x_i^2 + z_i^2 - x_m^2 - z_m^2, i = 1, \dots, m-1. \quad (2)$$

This system of  $m-1$  equations in 2 unknowns can be written as a matrix equation  $Ay=b$ , where  $A$  is the  $(m-1) \times 2$  matrix

$$A = 2 \cdot \begin{bmatrix} x_1 - x_m & z_1 - z_m \\ x_2 - x_m & z_2 - z_m \\ \vdots & \vdots \\ x_{m-1} - x_m & z_{m-1} - z_m \end{bmatrix}$$

$y^T = (x_c, z_c)$ , and  $b^T = (x_1^2 + z_1^2 - x_m^2 - z_m^2, \dots, x_{m-1}^2 + z_{m-1}^2 - x_m^2 - z_m^2)$ . If the rank of A is 2 then the least squares problem

$$Ay = b \quad (3)$$

has a unique solution  $\hat{y}$  that is determined by solving the normal equations

$$A^T Ay = A^T b. \quad (4)$$

The unique solution to (4) (if it exists) is  $\hat{y} = (A^T A)^{-1} A^T b$  and is simple to obtain since  $A^T A$  is a 2x2 matrix.

Finally,  $r$  can be obtained using (1) and any one of the  $m$  points, although an average of the values of  $r$  found with several points is preferred.

In order to determine the accuracy of this technique the sum of the deviations, in the  $z$  direction, from the circle at each point is computed.

Thus  $e = \sum_{i=1}^m (z_i - zp_i)$  where  $zp_i = z_c + \text{Round}((r^2 - (x_i - x_c)^2)^{1/2})$ . If  $e$  is small then the points of D lie on a circle with radius  $r$  and center  $(x_c, y_o, z_c)$ . The size of  $e$  which is acceptable is expected to depend on  $r$  or  $m$ .

The above algorithm should be repeated to locate several circles between the two parallel lines  $l_1$  and  $l_2$ , all having approximately the same radius and having their centers on a single line (the proposed axis of the cylinder) before concluding that a cylinder has actually been detected and localized.

Further analysis is needed to determine a priori error bounds for the solution of (4) because of the ill-conditioned problem that often results from forming the normal equations.

Given that this algorithm was tested on ideal data (rather than the noisy data from an actual sensor) the next step in this research effort is to test the robustness of the algorithm in the presence of noise. Future effort will also involve implementing this algorithm in the case when the cylinder is not in standard position but has a nonzero angle of inclination between its axis and the  $y$ -axis. The current algorithm represents a step toward an autonomous system for detection and localization of cylindrical objects in images created using range data.

## References

- Ballard, D.H., *Pattern Recognition*, 1981, 13(2), pp 111-122.
- Davis, E.R., *Pattern Recognition Letters*, 1987, 6, pp 1-7.
- Faugeras, O.D.; Herbert, M., *The International Journal of Robotics Research*, 1986, 5(3), pp 27-52.
- Gonzalez, Rafael C.; Wintz, Paul, *Digital Image Processing*, 1987, 2nd ed., Reading, Mass: Addison-Wesley, pp 331-349.
- Gorman, Frank and M.B. Clowes, *IEEE Transactions on Computers*, 1976, 25(4), pp 449-456.
- Kittler, J., *Image and Vision Computing*, 1983, 1(1), pp 37-42.